

MICROPOLIS BASIC 4.0 VARIABLE AND ARRAY STORAGE LOCATION

by Keith B. MacAdam

350 Blueberry Road, Lexington KY 40506

Tel. (606) 257-3344 (work), 277-1993 (home)

It might be useful to users of Micropolis BASIC 4.0 to know how and where BASIC stores variables, so that direct access to them can be had by assembly-language routines called from BASIC. I investigated this recently while writing routines to support a transient digitizer (Rev. Sci. Instrum. 52, 297 (1981)) and its oscilloscope display in my atomic physics laboratory. What follows is a description of the vector table, contained within the BASIC interpreter, that points to the storage locations for BASIC variables and arrays. Although the storage locations will change as a program is modified, the locations of the vectors remain fixed. Therefore assembly-language routines can look at the fixed vector locations and thus remain valid for different versions of a BASIC program. All vectors are stored in low-high format, that is, a hexadecimal memory address ABCDh will be found stored as CD AB in a memory dump. The first, or lower-address, location (containing the lower-order byte) will be given below as the vector locations.

VECTOR TABLE

331B	Vector to the first byte following the table	
331D	Vector to assembly-language function FAA	
331F	" " " " " "	FAB
33nn	" " " " " "	FAn
334F	" " " " " "	FAZ
3351	Vector to BASIC user-defined function FNA	
33nn	" " " " " "	FNn
3383	" " " " " "	FNZ
3385-33B8	Vectors to real arrays A(),...,Z().	(26x2 bytes)
33B9-33EC	Vectors to string arrays A\$(),...,Z\$().	(26x2 bytes)
33ED-3420	Vectors to integer arrays A%(),...,Z%().	(26x2 bytes)
3421	Vector to the start of the A0,...,Z0 alphabet	
3423	" " " " " " A1,...,Z1	"
34nn	" " " " " " An,...,Zn	"
3433	" " " " " " A9,...,Z9	"
3435	" " " " " " A,...,Z	"
3437-346A	Vectors to strings A\$,...,Z\$.	(26x2 bytes)
346B	Vector to the start of the A%,...,Z% alphabet	
346D-3480	Space for ten more vectors	(unknown)
3481	First byte after the vector table (pointed to by 331B)	

PIP will allow you to copy a file from one user area to another by adding an option to the PIP command line. For example, to copy the file MEMORITE.SYS from User 5 to User 6 on Drive A, enter User 6 (the destination User) with the command "USER 6<cr>".

Type the command -

```
"PIP A:=MEMORITE.SYS[G5]<cr>".
```

The [G5] tells PIP to get the file from User 5 and it copies the file into the default User Area (6).

A couple of quick points to remember: Using different user areas does not buy you more space or directory entries on your disk. Each file in each user area uses up free space and directory entries on your disk. Space is dynamically allocated when needed, so you can have all of your files on one user area (typically 0) or spread them out into many different user areas. The disk space penalty is the same.

Also, you can change default user areas in some application software packages. For instance, Memorite's command 'SU' for "Set User" will tell Memorite to get its Memorite files in the user area you specify. ExecuPlan has an equivalent command.

User areas can help you organize your life by organizing your computer files. Try it and see what happens.

MORE ON BASIC VECTOR POINTERS

by George Maschino

Rt 2 Box 24, Okarche OK 73762 - (405) 263-7614

The information in newsletter #68 by Keith Mac Adam is very good, and just what I wanted. My system does just as he described with a small exception. In the header for string arrays he shows a typical header of:

```
01 00 01 0C / 0A 05 data.....
```

My system shows a dump as;

```
01 00 01 00 0C / 00 0A 00 05 data....
```

It seems that each of the parameters in the string array header is a two byte value except the #dim.

On the 10 unknown vectors mentioned, 346D to 3480, they point to the FCB, which is the memory location describing the files that are open. When you open a file, Basic fills in the appropriate vector and the FCB for that file with information regarding that file. As an example I typed:

```
OPEN 0 "DINV"
OPEN 1 "1:INV"
OPEN 2 "2:ACCT.P"
```

```
OPEN 3 "2:FIX1"
OPEN 4 "*P":OPEN 5 "*T":OPEN 6 "*N"
LINK "MDOS"
```

By opening consecutive file #n, I could determine the end location of each FCB as also being the beginning of the next one. I dumped each FCB separately. See the dump at the end of this article. This made it easy to compare them.

The first byte in the FCB describes the device being addressed. For:

```
01 it is a disk file.
02 it is the printer, i.e., OPEN 4 "*P"
03 it is the terminal
00 it is the garbage can, i.e., OPEN 6 "*N"
```

I do not know the significance of the next 13 bytes. They change if you read or write to the file. Note that the 16th byte (last byte top row) is 04 in all disk files and 00 in other devices. This may be a device code but seems redundant with the first byte. All devices or files include the bytes BE FF. Why? The first seven bytes in the second row of each FCB are all 00.

Remember that I did not read or write to any of these files. I only opened them. If I had used a file (read or write to it), some or all of these would have been changed. The eighth byte in the second row is the disk drive #. The first file is from drive # 0. The second file is from drive #1 and the next two came from drive #2. The ninth byte in the second row is different for each file, but I do not know its significance.

The next 16 bytes in each FCB come from the directory of the disk with the file. The first ten bytes contain the filename. The eleventh byte is the track # of first track of the file, with the msb set. The twelfth byte is for file type. That is, the 93 in the first FCB is 13 hex. The file in the 4th FCB is a Basic program or a type 10 file. The thirteenth and fourteenth bytes are the record number (hex) having the end of file flag. In the top FCB that is record # 00 0D or 0C hex data records. The second FCB has a long file of 03 B7-1 records (hex). The fifteenth and sixteenth bytes came from the disk directory, but I do not know what they mean.

The last two bytes in the FCB are the file size in low-high sequence. I suspect that this is the put-seek pointer and the other pair is the actual filesize. The two bytes just preceeding these are 01 00 in all disk FCBs. Since I did not read from any of these files, the get-seek pointer will still point to record #1, and that is what these two bytes seem to do... 01 00 low-high order. The next pair, in the same order, pointing to the EOF flag, seem to be the logical put-seek pointer.

The last three FCBs are for printer, terminal and null files. The only difference between these three seem to be in the first byte and that is already described above. The sixteenth byte in these three FCBs are all the same

00 but different than the 04 for disk files. The AA in these three FCBs is probably a false value. I filled memory with AA so that I could tell if Basic changed things or if memory still held the garbage it powered up with. I think the AA in these three FCBs is garbage left when Basic did not use this byte for device files like it did for disk files.

Finding this block of vectors opens up possibilities. Now I can get to work on that ASSM language binary sort program that I had been thinking about. Thanks Keith.

Unknown Vectors Referred to by Keith Mac Adam

```
>DUMP 346D 3480
346D BB 5D E8 5D 15 5E 42 5E 6F 5E 9C 5E C9 5E 00 00 .].^B^o^o^o^
347D 00 00 00 00
```

Dump of FCBs

```
>DUMP 5DBB 5DE8
5DBB 01 00 00 00 00 00 00 BE FF 00 00 00 00 00 01 04 .....
5DCB 00 00 00 00 00 00 00 00 43 44 49 4E 56 20 20 20 .....CDINV

5DDB 20 20 20 93 00 00 0D 01 01 01 00 0D 00 .....
>DUMP 5DE8 5E15
5DE8 01 00 00 00 00 00 00 BE FF 00 00 00 00 00 01 04 .....
5DF8 00 00 00 00 00 00 00 01 32 49 4E 56 20 20 20 20 .....2INV
5E08 20 20 20 81 00 03 B7 01 01 01 00 B7 03 .....
>DUMP 5E15 5E42
5E15 01 00 00 00 00 00 00 BE FF 00 00 00 00 00 01 04 .....
5E25 00 00 00 00 00 00 00 02 36 41 43 43 54 2E 50 20 .....6ACCT.P

5E35 20 20 20 86 00 00 03 01 01 01 00 03 00 .....
>DUMP 5E42 5E6F
5E42 01 00 00 00 00 00 00 BE FF 00 00 00 00 00 01 04 .....
5E52 00 00 00 00 00 00 00 02 43 46 49 58 31 20 20 20 .....CFIX1
5E62 20 20 20 93 10 00 03 01 01 01 00 03 00 .....
>DUMP 5E6F 5E9C
5E6F 02 00 00 00 00 00 00 BE FF 00 00 00 00 00 01 00 .....
5E7F 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
5E8F 00 00 00 00 00 00 00 00 00 00 00 00 00 AA .....
>DUMP 5E9C 5EC9
5E9C 03 00 00 00 00 00 00 BE FF 00 00 00 00 00 01 00 .....
5EAC 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
5EBC 00 00 00 00 00 00 00 00 00 00 00 00 00 AA .....
>DUMP 5EC9 5EFF
5EC9 01 00 00 00 00 00 00 00 BE FF 00 00 00 00 00 01 00 .....
5ED9 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
5EE9 00 00 00 00 00 00 00 00 00 00 00 00 00 AA .....
```

Figure 1 - MDOS FCB Dumps (Maschino)

TIME MANAGEMENT AND SPELLBINDER

(Reprinted from Lexisoft Dealer Notes)

Blake Thomson probably doesn't know what he started (at least not until he reads this).

Thomson, of Systems Directions in Gregory, Michigan, recently told us he uses Spellbinder to keep track of things he needs to do. He used to do a list by hand, but with Spellbinder he spends "more time getting projects done than in working on the list."

He started by typing into his empty workspace a list of things to do, entering them as they occurred to him. He then used the h (Hold) Command to pick items one at a time, in order of priority, and put in the Hold Buffer.

"Once everything is in the hold buffer, simply use the u (Unhold) Command to print out the list, and save it for the next revision," Thomson says.

"Each day I bring up the list and review it, deleting the items that have been accomplished. Then I add new items or reverse existing items on the list. A rather simple idea, but it is a real timesaver for me, and hopefully for other Spellbinder users."

Making a List Work for You

We liked Thomson's idea, and it started us thinking of ways we might improve on it, especially if the list of projects is a long one.

If you assign each item a ranking, you can use the ALPHA Macro to put the list into priority order, instead of doing it by hand. For example, you might have a list something like this (only much longer):

- 2 Get sales figures from Sam
- 1 Order Spellbinder Brochures
- 3 Call Creative Media Services
- 1 Ask Maxine re demo
- 2 Follow through with Wilson account

"1" might mean "First Priority," or it might mean "Do This Week;" "2" might mean "Secondary Priority," or "Do Next Week," and so on. ALPHA will group the list like this:

- 1 Order Spellbinder Brochures
- 1 Ask Maxine re demo
- 2 Follow through with Wilson account
- 2 Get sales figures from Sam
- 3 Call Creative Media Services

If the number stands for the week, you can update your list each Friday by using Search and Replace to change all 2's ("Next Week") to 1's ("This Week").

A Project or Appointment Calendar

To create a schedule or calendar, start each line with a date, perhaps the one by which the project must be completed:

02-21 Get sales figures from Sam

In the case of an appointment calendar, add the time.

03-12 AM 08:45 Expect call from Brian.

You can then sort these using ALPHA. Just make sure you: 1) put in a zero before single digits as above (03-12 for March 12, for example); 2) use a hyphen and not a slash mark to separate digits (03-12, not 03/12); 3) put "AM" or "PM" before the time; and 4) answer "no" (0) when you're asked if you want a Validation Check.

Each morning ALPHA will generate a list of projects or appointments for that day. When you're asked for a "Minimum Target" and "Maximum Target" enter Today's date (for example, 02-21) for both. By the way, the original master list need not be in chronological order. For example, from a list like this:

```
02-21 AM 08:45 Do this
03-04 AM 11:00 Do this
01-31 PM 04:00 Do this
01-31 AM 11:00 Do this
02-21 AM 09:45 Do this
03-04 PM 04:00 Do this
02-21 AM 06:45 Do this
03-04 AM 10:00 Do this
02-21 AM 08:15 Do this
03-04 AM 07:00 Do this
02-21 AM 08:05 Do this
03-04 PM 02:30 Do this
```

ALPHA can generate a list for March 4 in hour-by-hour order:

```
03-04 AM 07:00 Do this
03-04 AM 10:00 Do this
03-04 AM 11:00 Do this
03-04 PM 02:30 Do this
03-04 PM 04:00 Do this
```

If you want an entire week's appointments or projects, give Monday's date (03-04) for the "Minimum Target" and Friday's date (03-08) for the "Maximum Target."

Spellbinder Makes It Possible

Such useful memory aids are proof positive of how powerful and flexible Spellbinder really is.

MICROPOLIS, TANDON AND MCI DRIVES

In previous newsletters we have let you know that we have parts for the Micropolis disk drives. We also have some Tandon and MPI drives. These were purchased from Vector Graphic at the time of the auction. Since we don't know very much about Tandon and MPI drives, we're selling them "as is", or you can order a specific part. These drives could come in handy since Tandon no longer makes the 100 TPI. Should your drive go down, it sure wouldn't hurt to have one sitting around to pull parts from. The same logic applies to the MPI, however I haven't the slightest idea whether the MPI is 100 TPI or not.

I'm selling the Micropolis drives for \$50, the Tandon's for \$75 and the MPI for \$50. These drives have full parts, but are untested, or, in some of the Micropolis cases, have tested bad.

I have some other drives that don't have all the parts. They may be missing a motor tach, or a bezel, etc. Call me if you need parts, and I'll pull them from the partial drives. Pricing would depend on what parts you required.

These are just a few of the parts available for the 1000 series and the 1115 series. Just because it's not on here doesn't mean I can't get it. If you have the 8" or the 5" hard disk I can also supply manuals and parts for them.

As mentioned last month - we have obtained the rights to distribute the source for MDOS and OSM. The cost is \$100 for MDOS and \$200 for OSM.

Micropolis Supplies

MANUALS

Maint Manual 1084-02	50.00
Maint Manual 1015	35.00
Maint Manual 1055	35.00
Maint Manual 1016F	35.00
Maint Manual 1115	50.00
Maint Manual 1117	50.00
Maint Manual 1300	50.00
Maint Manual 1200	100.00
MDOS Users Manual	50.00

COMMON PARTS

Align Disk SS	50.00
Align Disk DS	100.00
Diagnostic Disk	50.00
Head Pads	.50
Sanders Maglube	4.50
Belt	6.05
Scotch 3M disk SSDD	32.30
Scotch 3M disk DDDD	47.00

1000 MICROPOLIS PARTS

Motor Tach Assy	38.00
Switch D/O&W/P	6.60
Clamp Assembly	10.75
Solenoid Assembly	12.66
Switch, Track 0	9.30
Stepper Motor	60.00
Load Arm	1.01
Spring eject	.50
Spring, Carr'g assy	.50
Pin Dowel upper	.50
Pin Dowel lower	.50
LED Assembly	8.47
Mounting Nut	.50
Spindal Assembly	35.00
Res Net DIP	3.00
Res Net SIP	1.78
Header 5-pin	.50
Receiver Assembly	35.00
Hinge	2.70
Handle	1.62
Head/Carr'g assy	114.00
Latch	6.44
Spring Door Open	.50
Pulley	.82
Pivot	.50
Bezel	3.92
Platten Assembly	7.35
Stand-off	.76

1115 MICROPOLIS PARTS

Head Carr'g assy SS	73.00
Head Carr'g assy DS	135.00
Spring Platen	.50
Spring Pltn Clmp	.50
Front Carr'g spg	.50
Sensor Assy	5.20
Motor Tach Assy	25.00
Index Assy	12.00
Index LED	7.50
Jaw Lifter	10.06
File Protect	3.18
LED	1.44
Spindal Assy	37.32
Switch/door sensr	1.09
Track zero assy	14.25
Upper Arm	3.00
Clamp Assy	4.00
Stepper Motor	48.00
Pulley	3.23
Bezel	3.97

MICROPOLIS/VG USERS GROUP

Newsletter #69
April 1986

CLASSIFIED

The Classified Section of the MUG Newsletter is available to all users, free of charge. Just call or write us with a description of what hardware, software, information, etc., that you want to obtain, or dispose of.

FOR SALE - Vector Graphic Multi-user System 5032, 2 CRT Video Mindless Terminals, TI 810 Printer, Hayes Micro-Modem, Memorite (extended), Execuplan II, TIM III Data Base Manager, Connect (modem software), Peachtree Accounting General Ledger. \$4,500 or best offer. Would like a cash offer or trade of a parallel printer for the TI-810.

R. C. Vierzba, American Brokerage Corp, PO Box 5089, Helena MT 59604, 1-(800) 328-8913 or (406) 449-3870.

WANTED - Micropolis 1223-I 8-inch Rigid (Hard) drive, preferably used. Tom, Horizon Data Systems, (201) 686-7000.

FOR SALE - Vector Graphic 3 with a Micropolis 1053-II, MDOS, CP/M, Peachtree accounting software. \$795. Also a Qume Sprint II printer for \$1295, or best offer. Harold Cornine, (816) 532-0996

FOR SALE - Vector Graphic SX 2000 with 128K RAM, two quad-density, double-sided floppy drives. The condition is mint, and actual running time is estimated to be less than 500 hours. \$1800, shipping included.

Robert T. Martin, Martin & Martin, Ltd., 110 Broadbent Road, Wilmington DE 19801, (302) 475-2525.

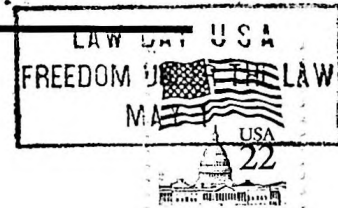
FOR SALE - Vector Graphic 3005, 5Mbyte hard disk, 640K floppy. Includes NEC 7700 printer, Peachtree accounting package, Chart, and ExecuPlan software. \$2000, with free delivery within 600 miles.

Robert Kaiser, 100 E. Washington, Monticello IL 61856, (217) 762-7815 (work) or (217) 762-3311 (home).

FOR SALE - TEI 12-slot mainframe with Cromemco Z80A CPU, 32K Dynabyte, 16K Problem solver, Cromemco 2S/2P, four Micropolis drives, MDOS, a Hazeltine 1500 terminal and a TI-810 printer. \$750 for all, or can break into parts.

Anne Meyer, Farmer's Pioneer Mutual Insurance Co., Onarga IL 60955, (815) 268-7300.

MICROPOLIS/VECTOR GRAPHIC
USERS GROUP
A Division of **DAMAN**
604 Springwood Circle
Huntsville AL 35803-1740
(205) 881-1697



FIRST CLASS MAIL

FIRST CLASS MAIL